Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computational and Applied Mathematics 195 (2006) 172–181
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Arc-length preserving curve deformation based on subdivision☆

Zhixun Su*, Ling Li, Xiaojie Zhou

Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, China

Received 15 August 2004; received in revised form 31 March 2005

Abstract

An arc-length preserving deformation for curves is presented by combining subdivision and inverse kinematic. A curve is discreted into polyline first, then the polyline is deformed with its arc-length preserved in the sense of minimizing energy. Subdivision method is applied to obtain a smooth curve (at least C^1 continuity) with proper weights selected to keep the length of the resulting curve equal to the original curve. This technique also provides interactive response by progressively refining the solution of the optimization problem.

Keywords: Deformation; Arc-length preserving; Subdivision

1. Introduction

Deformable modeling plays an important role in computer graphics. Typical examples are the animation of humans or animals, modeling of soft tissue in a virtual surgical training system, sewing up pieces of cloth in garment CAD and so on. During the past few decades, considerable work has been done in this area. In brief, the work can be divided into two parts [8]: non-physical models and physically based models. In general, non-physical models [2,19,1,13,16] are computationally efficient, but the results are less accurate than the physically based models and are limited by the experience of the user. In contrast, physically based models [21,18,4,12,7] give highly accurate deformation at the cost of computational complexity.

☆ This work is supported by the National Natural Science Foundation of China (no. 60275029).
* Corresponding author.
E-mail address: zxsu@comgi.com (Z. Su).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.03.092

Fig. 1. Algorithm overview.

Deformation with constraints is necessary to obtain realistic results. For example, cloth can only bear tiny stretch and shrink. Consequently, the local area of the cloth should be unchangeable when pieces of cloth are sewed up into a garment. Another example is that the skeleton length of an object usually keeps constant when it is axially deformed as stated in [16]. For instance, when an elephant curls its long nose, the length of its nose can be regarded as a constant. In 1980s, energy constraints [23], space time constraints and reaction constraints and augmented Lagrangian constraints [17] have been considered. Recently, Borrel and Rappoport [3] presented the volume-preserving deformation algorithm for free-form solids, in which the object was composed of several solids controlled by the lattice of tri-variate tensor product. A non-linear optimization was derived from the energy minimum to preserve the desired volume of each primitive and to ensure high-order continuity constraint between the primitives. Hirota et al. [11] presented an algorithm for preserving the total volume of an object undergoing free-form deformation using discrete level-of-detail representations. Aubert and Bechmann's method [1] was based on a space deformation model called DOGME. The volume was preserved and the algorithm was efficient, while the deformation depends too much on the optimization, the result was unpredictable. Also based on the DOGME model, Sun [20] modified the projection matrix and succeeded in preserving the area of 2D object. Peng et al. [16] presented an arc-length preserving axial deformation method. The axial curve was subdivided into polyline, then keyframe interpolation is used to the polyline with intrinsic definition to generate the arc-length preserving in-between frame. But this method is only suitable for morphing between the source curve and target curve. van Damme and Wang [22] presented a curve interpolation with constrained length which can preserve arc-length locally.

In this paper, we present an effective approach for arc-length preserving curve deformation based on subdivision. The arc-length preserving deformation for curves is implemented in three steps (shown in Fig. 1): First, control polyline of subdivision is obtained by simplifying the vast number of initial points (initial polyline) sampled from the source curve. Second, the arc-length preserving polyline deformation is obtained by solving a non-linear optimization based on inverse kinematics. At last, the polyline is subdivided to form the smooth deformed curve with proper weights selected to keep the length of the resulting curve equal to the original curve both locally and globally. Compared with [14] the computational cost is greatly reduced by integrating simplification technique and subdivision.

The rest of the paper is arranged as follows. Related work is introduced in Section 2. Section 3 explains how to obtain the control polyline. Section 4 describes how to deform the polyline with its length preserved. Finally, the polyline subdivision is described in Section 5. Experiments and results are presented in Section 6. Section 7 includes discussion and conclusion.

2. Related work

2.1. Curve and surface simplification

Many applications in computer graphics require complex, highly detailed models. For the sake of efficiency, it is often desirable to use approximations in place of excessively detailed models. Simplification is necessary to make storage, transmission, computation. It can reduce disk and memory requirements and speed up network transmission.

A variety of methods for simplifying curves and surfaces have been explored over these years. Heckbert and Garland [10] surveyed methods for simplifying curves and polygonal surfaces. The Douglas–Peucker algorithm is the most widely used high-quality curve simplification algorithm. It is a refinement algorithm. At each step, the algorithm attempts to approximate a sequence of points by a line segment from the first point to the last point. This algorithm produces the highest subjective- and objective-quality approximations. Ballard and Brown described a variant of the D–P algorithm. The algorithm splits at the point of highest error along the whole curve, instead of splitting recursively. This yields higher quality approximations for slightly more time. But they did not solve the problem of self-intersection. The algorithms for simplifying surfaces can be categorized into the classes below: vertex clustering, vertex decimation, iterative edge contraction, simplification envelopes, simplification through face removal, superface simplification, mesh optimization, re-tiling polygonal surfaces and multi-resolution analysis etc.

Enlightened by the face removal method of surface simplification, we choose the curvature to measure the importance of a vertex, and simplify the curves by deleting the vertices of less importance. And the curvature is computed by the directional angle of the vertices.

2.2. Curve and surface subdivision

As a new method for prescribing curves and surfaces, subdivision has been extensively studied in recent years. Compared with NURBS, subdivision has many advantages: it can generate surfaces with arbitrary topology and flexibly achieve shape characters by adjusting the weights in subdivision scheme, which is efficient and easy to implement since the new vertexes are calculated linearly and iteratively.

Subdivision curves are limit curves of a series of polylines, which are recursively computed using techniques such as "corner cutting". Chaikin first introduced the idea of subdivision in 1974, then Riesenfeld and Forrest proved that the limit curves are essentially uniform quadric B-splines. This algorithm generates a smooth curve that is approximately the shape of a user specified control polygon. The algorithm works by "clipping" the corners of the control polygon to form a new smoother control polygon. In fact, the limit curves generated by Chaikin's algorithm is proved to be uniform quadratic B-spline.

In 1987, Dyn et al. [6] pioneered a 4-point interpolatory subdivision scheme. It is a stationary scheme, but lacks of smoothness, only leading to C^1 curve. Cai [5] presented a 4-point subdivision with the tension parameter variable. Jin Jianrong and Wang Guozhao developed a non-uniform 4-point interpolatory subdivision scheme. G^1 interpolatory curve can be constructed by using this method. Some bias parameters are introduced to control the shape of curve. Cararetta, Dahmen and Micchelli presented a non-stationary subdivision scheme for interpolating a set of given data points in 1991, which is a generalization of the four point subdivision scheme to the non-stationary case. If the initial data lie on a $C^2(\mathbb{R})$ function, then the limit function of the scheme approximates the original function quadratically.

3. Simplification to control polyline

First, we sample the curve to obtain N source points $\{S_i\}_{i=0}^{N}$ in which the intrinsic definition of a planar curve is used [16], and then connect them orderly to form the initial polyline. We simplify it by deleting the points of "less importance".

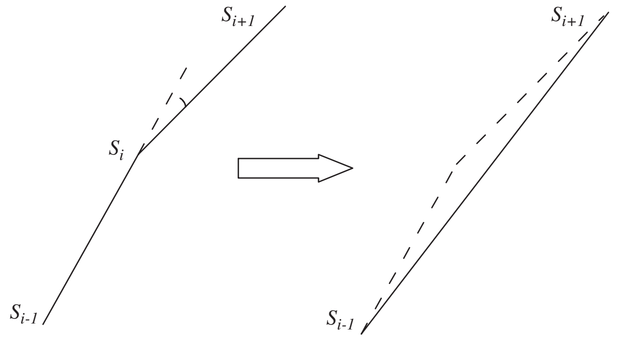

Fig. 2. Vertex decimate in data simplification.

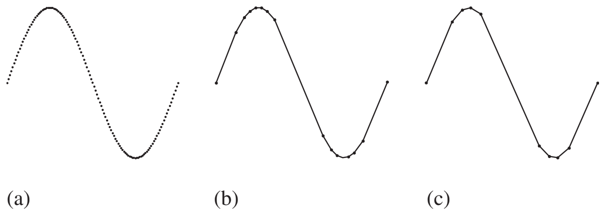

Fig. 3. Simplify the sinusoid curve. (a) Sampled sinusoid: 120 points. (b) Simplified sinusoid: 10°, 16 points. (c) Simplified sinusoid: 6°, 16 points.

The importance of a vertex is estimated by its directional angle δ_i (demonstrated in Fig. 2). If the absolute value of the angle is less than a given threshold δ, delete the point S_i, and record the lost arc-length. Finally, we get the rest points and connect them to form a new polyline, which we call control polyline (denoted by $\{P_i\}_{i=0}^{n}$). The other data obtained after simplification is the arc-length reduction $\{L_i\}_{i=1}^{n-1}$, where L_i is the arc-length lost during the deformation between control points P_i and P_{i+1}.

After simplification, the control polyline will reserve enough points in the high-detailed region and less data in the flat region. And the amount of the control points n depends on the threshold δ, the number of the control points decreases when the threshold increases (as shown in Fig. 3, the simplification of a sampled sinusoid curve using our algorithm).

4. Polyline deformation based on inverse kinematic chains

We have presented a deformation method of curves with the arc-length constraint in [14]. There are mainly three kinds of simple constraints for IK chains: (1) both the begin-point and the end-point are fixed; (2) the begin-point is fixed and end-point free; (3) both the begin-point and the end-point are free. More complicated constraints are syntheses of the simple cases. We carry out the deformation of polyline with the idea of inverse kinematic [9]. Inverse kinematic is a useful tool in robots programming and character animation. As user is specifying the target position of end-point (see Fig. 4), the computer will calculate the deformation result automatically. Based on it, the arc-length preserving deformation of curve with

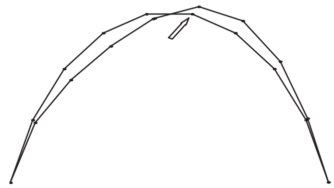

Fig. 4. Mid-point-goal deformation.

end-point moving is easy to achieve. But that is not enough for the requirements of deformation, we propose a deformation model suitable for case (1). Suppose that $\{P_i\}_{i=1}^{n}$ is the polyline with n vertexes, with both the begin-point and end-point fixed, move the mid-point P_m to the specified position X, and we will calculate the new polyline.

Calculate the edge lengths and the directional vertex angles of the polyline, and get its intrinsic definitions $\{\theta_0, (\theta_i, l_i)_{i=0}^{n-1}\}$. It is obvious that we can obtain the new positions of the vertexes through the directional vertex angles after deformation. So we calculate the shift of directional vertex angles $\Delta\theta = (\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_{m-1}, \Delta\theta_{m+1}, \ldots, \Delta\theta_n)$. At the point P_m, split the polyline into two parts, with each of begin-point fixed, and move the end-point to X. The left one is defined as $\{\phi_0, \phi_{\mathrm{L}} = (\phi_i, l_i)_{i=0}^{m-1}\}$, where $\theta_i = \phi_i$; and the right one $\{\phi_n, \phi_{\mathrm{R}} = (\phi_i, l_i)_{i=m+1}^{n-1}\}$, with $\theta_i = -\phi_i$. Then we have non-linear equations

$$f^{\mathrm{L}}(\theta_{\mathrm{L}}) = X,$$

$$f^{\mathrm{R}}(\theta_{\mathrm{R}}) = X, \tag{1}$$

where f^{L} and f^{R} are the positions of the end-points.

$$f^{\mathrm{L}}(\phi_1, \ldots, \phi_{m-1}) = \begin{pmatrix} x_1 + l_1 \cos\phi_1 + \cdots + l_{m-1}\cos(\phi_1 + \cdots + \phi_{m-1}) \\ y_1 + l_1 \sin\phi_1 + \cdots + l_{m-1}\sin(\phi_1 + \cdots + \phi_{m-1}) \end{pmatrix},$$

$$f^{\mathrm{R}}(\phi_{m+1}, \ldots, \phi_n) = \begin{pmatrix} x_n + l_{n-1} \cos\phi_n + \cdots + l_m\cos(\phi_n + \cdots + \phi_{m+1}) \\ y_n + l_{n-1} \sin\phi_n + \cdots + l_m\sin(\phi_n + \cdots + \phi_{m+1}) \end{pmatrix}.$$

Use the Taylor expansion, for $\phi_i + \Delta\phi_i \to \phi_i$, we have

$$J_{\mathrm{L}}\Delta\phi_{\mathrm{L}} = \Delta X,$$

$$J_{\mathrm{R}}\Delta\phi_{\mathrm{R}} = \Delta X, \tag{2}$$

where J_{L} is the Jacobian matrix of f^{L} with respect to ϕ_{L}, and J_{R} is the Jacobian matrix of f^{R} with respect to ϕ_{R}:

$$J_{\mathrm{L}} = \begin{pmatrix} J_{11} & J_{12} & \ldots & J_{1(m-1)} \\ J_{21} & J_{22} & \ldots & J_{2(m-1)} \end{pmatrix}, \quad J_{\mathrm{R}} = \begin{pmatrix} J_{1(m+1)} & J_{1(m+2)} & \ldots & J_{1n} \\ J_{2(m+2)} & J_{2(m+2)} & \ldots & J_{2n} \end{pmatrix}, \tag{3}$$

where

$$
\begin{aligned}
J_{11} &= -l_1 \sin\theta_1 - l_2 \sin(\theta_1+\theta_2) - \cdots - l_{m-1}\sin(\theta_1+\theta_2+\cdots+\theta_{m-1}),\\
J_{12} &= -l_2 \sin(\theta_1+\theta_2) - \cdots - l_{m-1}\sin(\theta_1+\theta_2+\cdots+\theta_{m-1}),\\
&\vdots\\
J_{1(m-1)} &= -l_{m-1}\sin(\theta_1+\theta_2+\cdots+\theta_{m-1}).
\end{aligned}
$$

$$
\begin{aligned}
J_{21} &= l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2) + \cdots + l_{m-1}\cos(\theta_1+\theta_2+\cdots+\theta_{m-1}),\\
J_{22} &= l_2 \cos(\theta_1+\theta_2) + \cdots + l_{m-1}\cos(\theta_1+\theta_2+\cdots+\theta_{m-1}),\\
&\vdots\\
J_{2(m-1)} &= l_{m-1}\cos(\theta_1+\theta_2+\cdots+\theta_{m-1}).
\end{aligned}
$$

$$
\begin{aligned}
J_{1(m+1)} &= l_m \sin(\theta_{m+1}+\cdots+\theta_n),\\
J_{1(m+2)} &= l_m \sin(\theta_{m+1}+\cdots+\theta_n) + l_{m+1}\sin(\theta_{m+2}+\cdots+\theta_n),\\
&\vdots\\
J_{1n} &= l_m \sin(\theta_{m+1}+\cdots+\theta_n) + l_{m+1}\sin(\theta_{m+2}+\cdots+\theta_n) + \cdots + l_{n-1}\sin\theta_n.
\end{aligned}
$$

$$
\begin{aligned}
J_{2(m+1)} &= l_m \cos(\theta_{m+1}+\cdots+\theta_n),\\
J_{2(m+2)} &= l_m \cos(\theta_{m+1}+\cdots+\theta_n) + l_{m+1}\cos(\theta_{m+2}+\cdots+\theta_n),\\
&\vdots\\
J_{2n} &= l_m \cos(\theta_{m+1}+\cdots+\theta_n) + l_{m+1}\cos(\theta_{m+2}+\cdots+\theta_n) + \cdots + l_{n-1}\cos\theta_n.
\end{aligned}
$$

Here Eq. (2) is a linear system of four equations of n variables. It is an underdetermined system. In order to get the unique solution, we have to define an objective function to solve the problem as an optimization problem. Here we consider the angle limits, that is, each θ_i should have upper bound and lower bound: $\theta_{i\,\min} \leqslant \theta_i \leqslant \theta_{i\,\max}$. It is based on some physical characters, for examples, the anti-bending rope, and joint limit of man's skeleton. We use a modified function proposed in [15]:

$$
L(\Delta\theta) = \frac{1}{n}\left(\sum_{i=1}^{n} w_i \left(\frac{(\theta_i - a_i) + \Delta\theta_i}{a_i - \theta_{i\,\max}}\right)^2\right). \tag{4}
$$

Optimizing this function ensures the angle limit, and minimizes the kinetic energy. And the weight w_i can be used to adjust the shape of the curve. At last, in order to limit the directional vertex angle at the vertex P_m, we require the end-point of the whole polyline $\{P_i\}_{i=1}^{n}$ satisfies the zero-displacement constraint,

$$
J\Delta\theta = 0, \tag{5}
$$

where $J = \{J_{pq}\}$ $(p = 1, 2;\ q = 1, \ldots, (n-1))$.

Thus the problem of solving the equation system Eq. (2) has been converted to an optimization problem:

$$
\begin{aligned}
\min \quad & L(\Delta\theta)\\
\text{s.t.} \quad & \mathbf{E}\Delta\theta = \mathbf{e},
\end{aligned} \tag{6}
$$

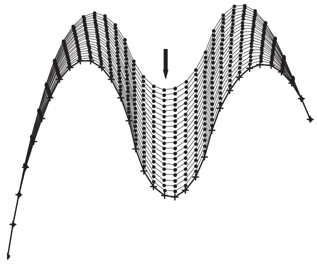

Fig. 5. Arc-length preserving curve deformation with two end-points fixed.

where

$$\mathbf{E}=\begin{pmatrix} J_{11} & \cdots & J_{1(m-1)} & 0 & 0 & \cdots & 0 & 0 \\ J_{21} & \cdots & J_{2(m-1)} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & J_{1(m+1)} & \cdots & J_{1(n-1)} & J_{1n} \\ 0 & \cdots & 0 & 0 & J_{2(m+1)} & \cdots & J_{2(n-1)} & J_{2n} \\ J_{11} & \cdots & J_{1(m-1)} & J_{1m} & J_{1(m+1)} & \cdots & J_{1(n-1)} & J_{1n} \\ J_{21} & \cdots & J_{2(m-1)} & J_{1m} & J_{2(m+1)} & \cdots & J_{2(n-1)} & J_{2n} \end{pmatrix}, \quad \mathbf{e}=\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta x \\ \Delta y \\ 0 \\ 0 \end{pmatrix}.$$

It is an optimization problem with six linear equality constraints. We can solve it by gradient project or feasible direction method, and the optimal angle shift is obtained. Then we can calculate the new vertexes to obtain the deformed polyline. It has been implemented on PC (as shown in Fig. 5).

Using the above algorithm, the total arc-length, even the arc-length between any two control points is preserved. However, the algorithm is not suitable for curve deformation with large numbers of points. If the curve is complex and highly detailed, direct operation on source points is costly, and is not always required. Here we apply it to deform the polyline, and ensure the total length and segment length of control polyline to remain constant at the same time.

5. Subdivision with arc-length preserved

We apply subdivision to the deformed control polyline obtained in Section 4. The subdivision scheme is chosen from Cai [5], which is a 4-point subdivision with the tension parameter variable. The subdivision scheme defines the control points at level $k+1$ by

$$\begin{aligned} &P_{2i}^{k+1} = P_i^k, \quad -1 \leqslant i \leqslant 2^k n + 1, \\ &P_{2i+1}^{k+1} = (\tfrac{1}{2} + w_i^k)(P_i^k + P_{i+1}^k) - w_i^k(P_i^k + P_{i+1}^k), \quad -1 \leqslant i \leqslant 2^k n. \end{aligned}$$

It is also proved that the scheme produces C^1 curves while $0 < \alpha \leqslant w \leqslant \beta < \frac{1}{8}$.

Notice that during the subdivision process, the arc-length of the curve increases monotonously as the subdivision level k increases. What's more, for a given k, the arc-length increases monotonously as w increases. In order to preserve the arc-length, we adjust the tension parameters w_i so that the arc-length increased in subdivision will be equal to the lost in simplification.

We apply dichotomizing search to seek the suitable w_i for subdivision between $P_i P_{i+1}$ as follows: let $f(w)(w \in [0, \frac{1}{10}])$ be the function that defines the arc-length between $P_i P_{i+1}$ after subdivision.

(1) Initialize the subdivision level k, $w_i = \frac{1}{16}$;
(2) Subdivide k levels, and compute the sum of length between P_{i-1} to P_i, seek a w_i in $[1, \frac{1}{10}]$ to satisfy $f(w_i) = \overline{P_{i-1}P_i} + L_i$. If $f(\frac{1}{10}) < \overline{P_{i-1}P_i} + L_i$, return to the simplification to readjust the angle threshold;
(3) Execute (2) until the inequality $|f(w_i) - (\overline{P_{i-1}P_i} + L_i)| < \varepsilon$ is satisfied, where ε is a given tolerance.

6. Experiments and results

We have implemented our algorithm on Intel Pentium 4 CPU 1.70 GHz PC. The result of a deformed circle is shown in Fig. 6. The circle is sampled into 120 points and simplified into 35-point control polyline, and cost 51 centiseconds aggregately. The open curve is sampled into 120 points and simplified into 28 points, and cost 47 centiseconds aggregately (Fig. 7). We observe that if setting a too big angle threshold in simplification, the subdivided curve will be lack of fairness. For example, the circle

Fig. 6. The circle before and after deformation.

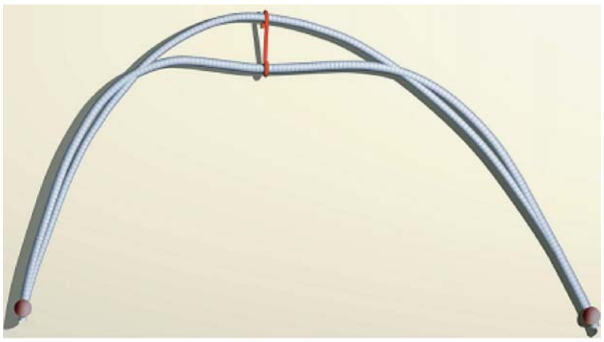

Fig. 7. The open curve before and after deformation.

Fig. 8. The deformed circle lack of fairness.

(120 sampling points) will be deformed into the shape shown in Fig. 8 when using a threshold of 18 degree (simplified from 120 points to 11 points).

7. Discussion and conclusion

The present method achieves arc-length preserving deformation for curves by combining subdivision and inverse kinematic. The method preserves not only the total length of the curve, but also the arc-length between each two adjacent control points. Four-point scheme is an interpolatory subdivision, and only leads to C^1 curves. Therefore the result cannot always maintain fairness under the constraint of interpolation. We expect to improve our algorithm to get smooth curves. Future work should focus on how to select the subdivision scheme and the simplification threshold. Only curve deformation is discussed in this paper, surface deformation with constraints should be considered in the future.

References

[1] F. Aubert, D. Bechmann, Volume-preserving space deformation, Comput. Graphics 21 (1997) 625–639.
[2] A.H. Barr, Global and local deformation of solid primitives, Comput. Graphics 18 (1984) 21–34.
[3] P. Borrel, A. Rappoport, Simple constrained deformations for geometric modeling and interactive design, ACM Trans. Graphics 13 (1994) 137–155.
[4] D.E. Breen, D.H. House, M.J. Wozny, Predicting the drape of woven cloth using interacting particles, Comput. Graphics 28 (1994) 365–372.
[5] Z.J. Cai, Theory and application of 4-point subdivision scheme with variable parameter, Math. Annual 16A (1995) 524–531.
[6] N. Dyn, D. Levin, J. Gregory, A four-point interpolatory subdivision scheme for curve design, Comput. Aided Geom. Design 4 (1987) 257–268.
[7] O. Etzmuss, J. Gross, W. Strasser, Deriving a particle system from continuum mechanics for the animation of deformable objects, IEEE Trans. Visualization Comput. Graphics 9 (2003) 538–550.
[8] S.F.F. Gibson, B. Mirtich, A survey of deformable modeling in computer graphics, TR-97-19, Mitsubishi Electric Research Laboratory, Cambridge, MA, USA, 1997.
[9] M. Girard, A.A. Maciejewski, Computational modeling for the computer animation of legged figures, Comput. Graphics 19 (1985) 263–270.
[10] P.S. Heckbert, M. Garland, Survey of polygonal surface simplification algorithms, in: Proceedings of the SIGGRAPH'97, Multiresolution Surface Modeling Course Notes, 1997.

[11] G. Hirota, R. Maheshwari, M.C. Lin, Fast volume-preserving free form deformation using multi-level optimization, Comput. Aided Design 32 (2000) 499–512.
[12] D.L. James, D.K. Pai, ArtDefo: accurate real time deformable objects, Comput. Graphics 33 (1999) 65–72.
[13] F. Lazarus, S. Coquillart, E. Jancene, Axial deformations: an intuitive deformation technique, Comput. Aided Design 26 (1994) 607–613.
[14] L. Li, Z.X. Su, X.P. Liu, Application of rigid joint chain in arc-length-preserving curve deformation, J. Comput. Aided Design Comput. Graphics 17 (2005) 1002–1007.
[15] A. Liegeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Trans. System Man Cybern. 7 (1997) 869–871.
[16] Q.S. Peng, X.G. Jin, J.Q. Feng, Arc-length-based axial deformation and length preserving deformation, in: Computer Animation97, Geneva, IEEE Computer Society, Silver Spring, MD, 1997, pp. 86–92.
[17] J. Platt, A. Barr, Constraint methods for flexible models, Comput. Graphics 22 (1988) 279–288.
[18] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, in: Proceedings of Graphics Interface '95, 1995, pp. 147–154.
[19] T.W. Sederberg, R. Parry, Free-form deformation of solid geometric models, Comput. Graphics 20 (1986) 151–160.
[20] Y.S. Sun, Space deformation with geometric constraint, M.S. Thesis, Department of Applied Mathematics, Dalian University of Technology, 2002.
[21] D. Terzopoulos, D.J. Platt, A. Barr, K. Fleisher, Elastically deformable models, Comput. Graphics 21 (1987) 205–214.
[22] R. van Damme, R.H. Wang, Curve interpolation with constrained length, Computing 54 (1995) 69–81.
[23] A. Witkin, K. Fleischer, A. Barr, Energy constraints on parameterized models, Comput. Graphics 21 (1987) 225–232.